

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Автоматизованих систем обробки інформації і управління

(повна назва кафедри)

«До захисту допущено»

В.о. завідувача кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ ” 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки _____ **6.050103 «Програмна інженерія»**

на тему _____ *Програмне забезпечення формування рекомендацій з розміщення заходів*

Виконав:

студент IV курсу, групи ІП-52 Якимчук Олександр Анатолійович _____
(прізвище, ім'я, по батькові) (підпис)

Керівник

ст. викладач, Олійник Ю.О. _____
посада, науковий ступінь, вчене звання, прізвище, ініціали (підпис)

**Консультант
з графічної
документації**

доц., к.т.н., Ліщук К.І. _____
посада, науковий ступінь, вчене звання, прізвище, ініціали (підпис)

Рецензент:

доц. каф. ТК, к.т.н., доц.Пасько В.П. _____
посада, науковий ступінь, вчене звання, прізвище, ініціали (підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЕКТУ

[illegible]

АНОТАЦІЯ

Робота містить 11 рисунків та 22 таблиці.

Важко уявити сучасний Інтернет без рекомендаційних систем. Вони обробляють величезні об'єми інформації та видають персоналізований результат який найкраще відповідає нашим смакам та інтересам. Допомагають нам не загубитись у нескінченних потоках інформації та знайти те, що нам потрібно у найкоротший термін.

З іншого боку, ці системи також є джерелом прибутку для підприємств торгівлі що володіють ними. Адже вони допомагають завоювати довіру і лояльність користувачів.

Зокрема це стосується соціальних мереж, що базуються на проведенні заходів. Саме на допомогу в останній сфері направлене дане програмне забезпечення. Проаналізувавши смаки на побажання користувачів на основі наданої клієнтської бази, система поверне список всіх потенційних відвідувачів з необхідною інформацією. Це допоможе організатору краще зрозуміти цільову аудиторію та побудувати ефективну рекламну кампанію.

КЛЮЧОВІ СЛОВА: РЕКОМЕНДАЦІЙНА СИСТЕМА, РЕКЛАМА, РОЗМІЩЕННЯ ЗАХОДІВ

ABSTRACT

The work contains 11 drawings and 22 tables.

It's hard to image a modern Internet without a system of recommendations. They process a huge amount of information and provide a personalized result that best suits our tastes and interests. Help us not get lost in endless streams of information and find what we need as quickly as possible.

On the other hand, these systems are also a source of profit for the trading companies that own them. After all, they help to win the trust and loyalty of users.

In particular, this applies to social networks based on activities undertaken. It was for the help in the latest area that this software was sent. Having analyzed the tastes according to the wishes of users on the basis of the provided client base, the system will return the list of all potential visitors with the necessary information. This will help the organizer to better understand the target audience and build an effective advertising campaign.

KEYWORDS: RECOMMENDATION SYSTEM, ADVERTISING, EVENT HOSTING

Пояснювальна записка до дипломного проекту

на тему: Програмне забезпечення надання рекомендацій з розміщення заходів

Київ – 2019 року

					КПІ.ІП-XXXX.XXXXXXX.XX.XX	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	10
ВСТУП.....	11
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	12
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ.....	15
1.3.1 Аналіз відомих технічних рішень	15
1.3.2 Аналіз відомих програмних продуктів.....	16
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	17
1.4.1 Розроблення функціональних вимог.....	18
1.4.2 Розроблення нефункціональних вимог	24
1.4.3 Постановка комплексу завдань модулю	24
1.5 Висновки по розділу	25
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	26
2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	26
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	27
2.3 ВХІДНІ ТА ВИХІДНІ ДАНІ.....	33
2.4 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	36
2.5 АНАЛІЗ БЕЗПЕКИ ДАНИХ.....	40
2.6 Висновки по розділу	40
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	41
3.1 АНАЛІЗ ЯКОСТІ ПЗ.....	41
3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....	52
3.3 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ	52
4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	56
4.1 РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	56
4.2 РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	57
ВИСНОВКИ	59

ПЕРЕЛІК ПОСИЛАНЬ.....60

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

API - Application Programming Interface

ПК – Персональний комп’ютер

ПЗ – Програмне забезпечення

MVT – Model-View-Template

ІТ – Інформаційні технології

БД – База даних

ВСТУП

Важко уявити сучасний Інтернет без рекомендаційних систем. Вони обробляють величезні об'єми інформації та видають персоналізований результат який найкраще відповідає нашим смакам та інтересам. Допмагають нам не загубитись у нескінченних потоках інформації та знайти те, що нам потрібно у найкоротший термін.

З іншого боку, ці системи також є джерелом прибутку для підприємств торгівлі що володіють ними. Адже вони допомагають завоювати довіру і лояльність користувачів.

Зокрема це стосується соціальних мереж, що базуються на проведенні заходів. Саме на допомогу в останній сфері направлене дане програмне забезпечення. Проаналізувавши смаки на побажання користувачів на основі наданої клієнтської бази, система поверне список всіх потенційних відвідувачів з необхідною інформацією. Це допоможе організатору краще зрозуміти цільову аудиторію та побудувати ефективну рекламну кампанію.

Актуальність теми – в наш час доволі важко не загубитись в величезних об'ємах інформації. Програмне забезпечення частково вирішить цю проблему для організаторів заходів.

Мета – покращення надання рекомендацій при створенні та просуванні заходів.

Призначення – отримання рекомендацій, що допоможуть ефективніше провести рекламну кампанію при просуванні заходу.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Рекомендаційна система — підклас системи фільтрації інформації, яка будує список об'єктів (фільми, музика, книги, новини, веб-сайти), яким користувач може надати перевагу. Для цього використовується інформація з профілю користувача.[1]

Кластеризація (або кластерний аналіз) - це задача розбиття множини об'єктів на групи, які називаються кластерами. Усередині кожної групи повинні виявитися «схожі» об'єкти, а об'єкти різних групи повинні бути якомога більш відмінні. Головна відмінність кластеризації від класифікації полягає в тому, що перелік груп чітко не заданий і визначається в процесі роботи алгоритму.[3]

Класифікація - задача розподілення об'єктів на групи залежно від наперед визначених ознак. В деяких випадках, вживають термін категоризація у значенні розподілення об'єктів на категорії.[3]

1.2 Змістовний опис і аналіз предметної області

Існують дві основні стратегії створення рекомендаційних систем: фільтрація вмісту і колаборативна фільтрація.

Фільтрація на основі вмісту, також відома як когнітивна фільтрація, рекомендує елементи на основі порівняння вмісту елементів і профілю користувача. Вміст кожного елемента представляється у вигляді набору термінів, зазвичай слів, що зустрічаються в документі. Профілі користувачів можуть містити демографічну інформацію або відповіді на певний набір питань.[1]

При колаборативній фільтрації за основу береться поведінка користувачів в минулому. Основне припущення полягає в тому, що користувачі які погодились в минулому схильні погоджуватись і в майбутньому.

					КПІ.ІП-XXXX.XXXXXXX.XX.XX	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

В цьому разі не має значення, з якими типами об'єктів ведеться робота, але при цьому можна брати до уваги неявні характеристики, які складно було б врахувати при створенні профілю. Основна проблема цього типу рекомендаційних систем — «холодний старт»: відсутність даних про користувачів чи об'єкти, які нещодавно з'явилися у системі.

Рекомендація заходів суттєво відрізняється від рекомендації книг та фільмів. Основні проблеми пов'язані з відсутністю даних про новий захід — проблема холодного старту. Книги і фільми вже переглянуті і оцінені багатьма користувачами.

На відмінну від них захід має чіткі часові рамки. Неможливо отримати відгук користувача про захід до його проведення, водночас, рекомендація заходу після його проведення не має ніякого сенсу.

При розробці програмного забезпечення я слідував гібридному підходу, який включає в себе фільтрацію вмісту та колаборативну фільтрацію.

Однією з основних проблем є величезна кількість представлених даних. Їх попередня обробка і витяг параметрів є критично важливим кроком, так як ми повинні отримати найбільш важливе з цього великого обсягу даних.

Для навчання моделі я обрав наступні параметри:

- а) тип заходу. Це один з найважливіших параметрів, адже від нього напряду залежить цільова аудиторія учасників;
- б) загальна кількість відвідувачів. Показник популярності заходу;
- в) кількість друзів, які відвідають захід. Ще один важливий параметр, оскільки друзі якнайкраще можуть підштовхнути користувача до прийняття рішення;
- г) відсоток друзів які відвідають захід. Якщо майже всі з друзів зацікавлені, вірогідно зацікавиться і користувач;
- д) місце проведення. Важливою є також географічна складова, а саме: чи співпадає країна користувача з країною в якій проводитиметься захід і чи

співпадає місто. Адже користувач з більшою вірогідністю прийме позитивне рішення, якщо для відвідування заходу йому не доведеться далеко їхати;

е) Вік та вікова група користувача. Заходи різних типів орієнтовані на різні вікові групи. Наприклад середньостатистичний користувач віком 19 років вірогідніше обере вечірку ніж театр;

ж) Стать;

з) День тижня;

и) Тривалість заходу;

к) Як часто користувач відвідує заходи. Якщо користувач часто відвідує різні заходи, він з більшою ймовірністю зацікавиться і в цьому;

л) Інтереси користувача. Як часто користувач відвідує заходи цього типу;

Перелічені вище параметри були здобуті після опрацювання наявних даних та збереженні до файлу `feature_train.csv`.

Для розбиття заходів за типом я використав ітераційний алгоритм Kmeans.

Алгоритм **Kmeans** намагається розбити заданий набір даних на K окремих підгруп(кластерів), що не перекриваються між собою. Кожна точка даних належить тільки одній групі. Алгоритм намагається зробити так, щоб міжкластерні точки даних були якомога ближчими, одночасно зберігаючи кластери якомога більш віддаленими один від одного. Він призначає точки даних кластерів таким чином, що сума квадрата відстані між точками даних і центроїдом кластера (середнє арифметичне всіх точок даних, які належать до цього кластера) є мінімальною. Чим менше варіації ми маємо в кластерах, тим більш однорідні (аналогічні) точки даних знаходяться в одному кластері.

1.3 Аналіз успішних ІТ-проектів

1.3.1 Аналіз відомих технічних рішень

Наступним важливим кроком після здобуття параметрів навчання став вибір алгоритму. При розробці програмного забезпечення я звернув увагу на наступні алгоритми:

– **Random forest** (з англ. - «випадковий ліс») – алгоритм машинного навчання, запропонований Лео Брейманом і Адель Катлер, що полягає у використанні комітету (ансамблю) дерев рішень. Алгоритм поєднує в собі дві основні ідеї: метод бегінга Брейман, і метод випадкових підпросторів. Алгоритм застосовується для задач класифікації, регресії і кластеризації. Основна ідея полягає в використанні великого ансамблю дерев рішень, кожне з яких саме по собі дає дуже невисоку якість класифікації, але за рахунок їх великої кількості результат виходить хорошим;[2]

– **Stochastic Gradient Descent (SGD)** – це простий, але дуже ефективний підхід до диференціального навчання лінійних класифікаторів при використанні функцій опуклих втрат, таких як лінійні машини опорних векторів і логічна регресія. Йому приділяється значна увага в контексті широкомасштабного навчання. SGD був успішно застосований до великомасштабних і рідкісних проблем машинного навчання, які часто зустрічаються в класифікації тексту та обробці природної мови;[3]

– **Наївний класифікатор Баєса** – простий імовірнісний класифікатор, заснований на застосуванні теореми Баєса зі строгими (наївними) припущеннями про незалежність. Залежно від точної природи ймовірнісної моделі, наївні баєсовські класифікатори можуть навчатися дуже ефективно. Незважаючи на наївний вигляд і, безсумнівно, дуже спрощені умови, наївні баєсовські класифікатори часто працюють набагато краще в багатьох складних життєвих ситуаціях. Перевагою наївного класифікатора Баєса є мала кількість даних необхідних для навчання, оцінки параметрів і класифікації;[6]

– **Класифікатор на основі дерева рішень** – при навчанні класифікуючої моделі використовується дерево рішень. Це один з підходів прогнозного моделювання, використовуваних в статистиці, інтелектуальному аналізі даних і машинному навчанні. Деревовидні моделі, в яких цільова змінна може приймати дискретний набір значень, називаються деревами класифікації; в цих деревовидних структурах листя представляють мітки класів, а гілки представляють з'єднання функцій, які ведуть до цих мітках класів. Деревя рішень, в яких цільова змінна може приймати безперервні значення (зазвичай дійсні числа), називаються деревами регресії;[7]

– **Extreme Gradient Boosting (xgBost)** – реалізує алгоритми машинного навчання за рамками градієнтного підсилення. XGBoost забезпечує паралельне підсилення дерев (також відоме як GBDT, GBM), яке дозволяє швидко і точно вирішити багато проблем науки науки. Той же код працює на великих розподілених середовищах (Hadoop, SGE, MPI) і може вирішувати проблеми поза мільярдами прикладів;

Після проведення тестів на тренувальному наборі даних я зупинився на одному алгоритмі, що показав найкращі результати: xgBoost.[5]

1.3.2 Аналіз відомих програмних продуктів

Рекомендаційні системи використовуються всюди, де є великі об'єми інформації. Прикладом тому можуть служити такі гіганти ринку як Google, YouTube, Instagram, Facebook. Всі вони використовують надзвичайно складні рекомендаційні алгоритми для поліпшення результатів.

1.4 Аналіз вимог до програмного забезпечення

Перш за все необхідно виділити ролі учасників.

– організатор заходу який хоче отримати інформацію про потенційних учасників.

Програмне забезпечення повинне містити в собі наступні можливості:

Для організатора:

- авторизація
- перегляд власних заходів
- редагування списку заходів(додання заходу, видалення заходу, редагування заходу)
- отримання інформації про потенційних учасників

В рамках дипломної роботи було детально розроблено можливості пов'язані з отриманням інформації про потенційних учасників. Інші вимоги стосуються білетного сервісу і будуть реалізовані там.

Детальні зв'язки між акторами та функціональними модулями програмного забезпечення вказані на діаграмі варіантів використання (Рисунок 1.1).

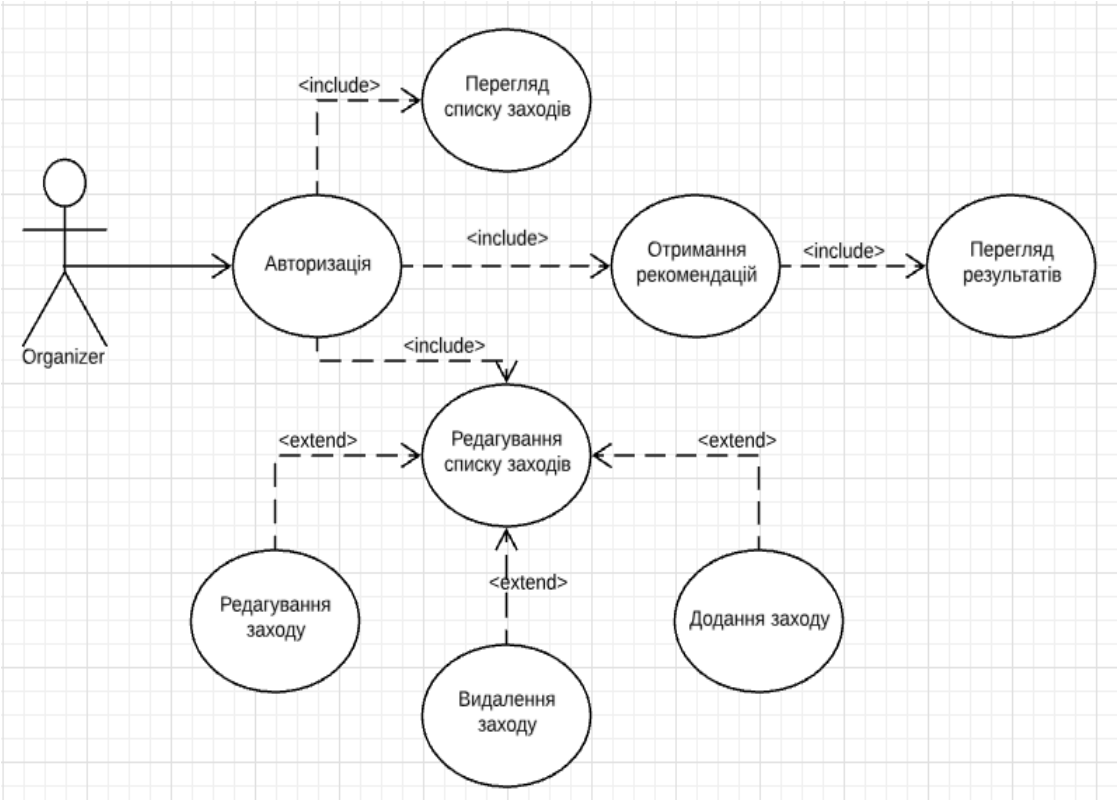


Рисунок 1.1. – Структурна схема варіантів використання

1.4.1 Розроблення функціональних вимог

Функціональні вимоги до програмного забезпечення наведені у таблиці нижче (Таблиця 1.1.).

Таблиця 1.1 – Функціональні вимоги

Актор	Варіант використання	Функціональна вимога	Пріоритет
Організатор	Авторизація	Необхідно зайти на сторінку авторизації і ввести логін та пароль у відповідну форму. Після чого натиснути кнопку «Вхід», що знаходиться внизу під формою	Високий
Організатор	Перегляд заходів	Перейти на сторінку «Мої заходи». На сторінці повинні відобразитись всі заходи користувача(назва та опис), де він є організатором.	Середній

Продовження таблиці 1.1

Організатор	Додання заходу	На сторінці «Мої заходи» натиснути кнопку «Додати захід», що знаходиться внизу сторінки. Ввести валіну інформацію про захід. Створити типи білетів та купони за необхідністю. Натиснути кнопку «Створити захід». Захід зберігається до БД. І при наступному відвідуванні сторінки «Мої заходи» він з'явиться у списку і буде доступний для редагування	Високий
-------------	----------------	--	---------

Продовження таблиці 1.1

Організатор	Видалення заходу	На сторінці «Мої заходи» навести мишкою на захід, який треба видалити та натиснути кнопку «видалити». Підтвердити дію в впливаючому вікні натиснувши кнопку «Підтверджую». Захід видаляється з БД. При наступному вході на сторінку «Мої заходи» захід не відображається у списку	Високий
-------------	------------------	---	---------

Продовження таблиці 1.1

Організатор	Редагування заходу	На сторінці «Мої заходи» обрати захід який треба редагувати на натиснути кнопку «редагувати». Ввести необхідну інформацію в форму. Натиснути кнопку «Зберегти». Зміни про захід зберігаються в БД	Високий
Організатор	Отримання списку потенційних клієнтів	На сторінці «Мої заходи» навести мишкою на захід, натиснути кнопку «Отримати рекомендації». Система формує рекомендації у вигляді списку потенційних відвідувачів та повертає їх у вигляді csv файлу	Високий

Матриця трасування вимог зображена на рисунку нижче (рисунок 1.2)

	Requirement: Авторизація	Requirement: Отримання рекомендацій	Requirement: Редагування списку заходів	Requirement: Перегляд власних заходів
UseCase Model: Авторизація	+	+	+	+
UseCase Model: Перегляд заходів				+
UseCase Model: Отримання рекомендацій		+		
UseCase Model: Перегляд результатів		+		
UseCase Model: Додання заходу			+	
UseCase Model: Перегляд результатів			+	
UseCase Model: Перегляд результатів			+	

Рисунок 1.2 – Матриця трасування

1.4.2 Розроблення нефункціональних вимог

Для використання рекомендаційної системи необхідно мати девайс, підключений до мережі Інтернет з встановленим веб-браузером.

Пропускна швидкість з'єднання повинна бути не менше 0.5 mb/s для комфортної роботи.

1.4.3 Постановка комплексу завдань модулю

Мета – покращення надання рекомендацій при створенні та просуванні заходів.

Призначення – отримання рекомендацій, що допоможуть ефективніше провести рекламну кампанію при просуванні заходу.

Задачі

- авторизація. Користувач вказує логін та пароль для входу в аккаунт. Система перевіряє ведені дані і, в залежності від результату перевірки, надає чи не надає користувачу доступ;

- перегляд заходів. Після авторизації користувач потрапляє на сторінку зі списком заходів, організатором яких він є;

- редагування списку заходів . Користувач має можливість додавати новий захід, видаляти та редагувати вже існуючі. Дана вимога не є частиною дипломної роботи і реалізована в білетному сервісі;

- отримання рекомендацій. Користувач повинен мати можливість отримати csv файл з інформацією про потенційних відвідувачів заходу.

1.5 Висновки по розділу

В даному розділі було описано загальні положення та предметну область. Проведений аналіз технічних рішень та відомих успішних ІТ проектів. Та сформовано перелік функціональних та нефункціональних вимог до програмного забезпечення.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Одним з ключових етапів в створенні програмного забезпечення є побудова діаграми бізнес-процесів. Для формування діаграми я обрав нотацію IDEF0 оскільки вона досить проста і зручна. А також я маю досвід роботи з нею.

Було розроблено 2 рівня схеми бізнес-процесів:

- 1) схема на рівні Системи в цілому (рисунок 2.1)
- 2) схема на рівні ключових модулів (рисунок 2.2)

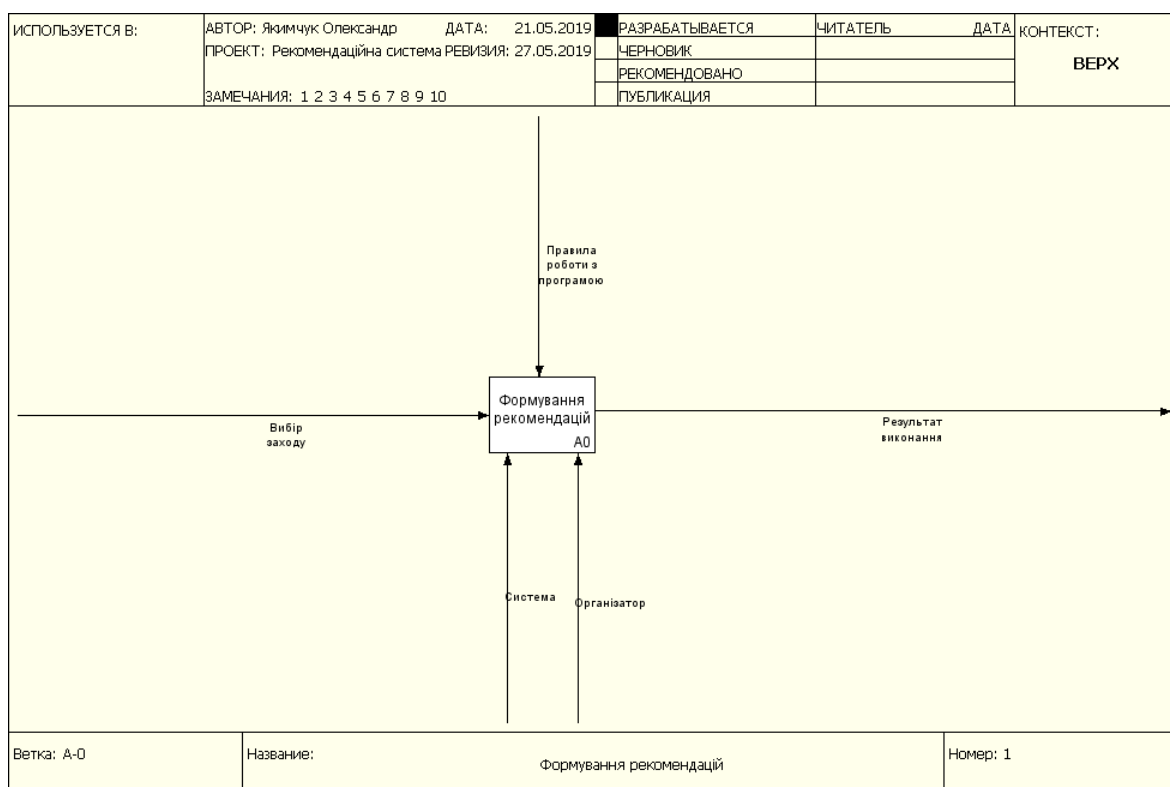


Рисунок 2.1 – Схема бізнес-процесу. Рівень 1.

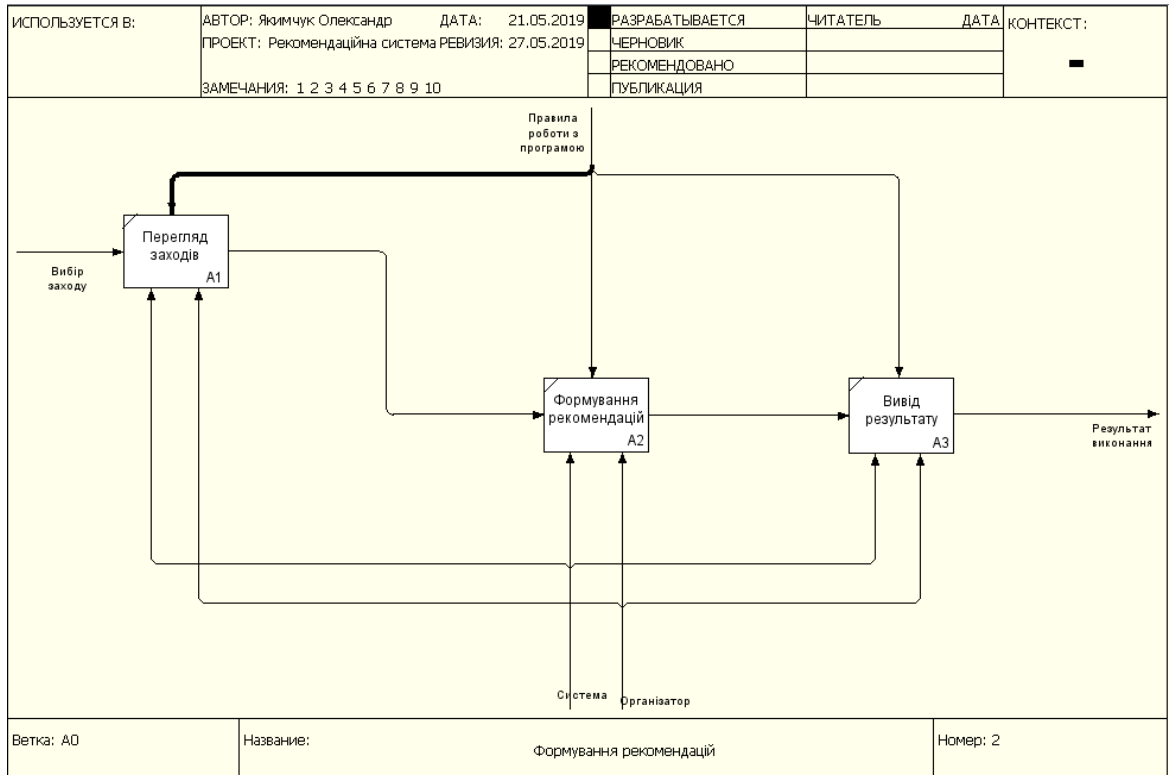


Рисунок 2.2 – Схема бізнес-процесу. Рівень 2

2.2 Архітектура програмного забезпечення

Для написання даного програмного забезпечення було обрано мову програмування Python версії 3.7.

В якості фреймворку я обрав Django версії 2.1. Це один з найпопулярніших фреймворків для написання синхронних веб-додатків. Містить багато вбудованих модулів та бібліотек, що полегшують розробку. Зокрема модуль авторизації, модуль для адміністрування бази даних, модуль для підтримки специфікації Rest API. В основі будь-якого Django додатку лежить архітектура MVT (Рисунок 2.3).

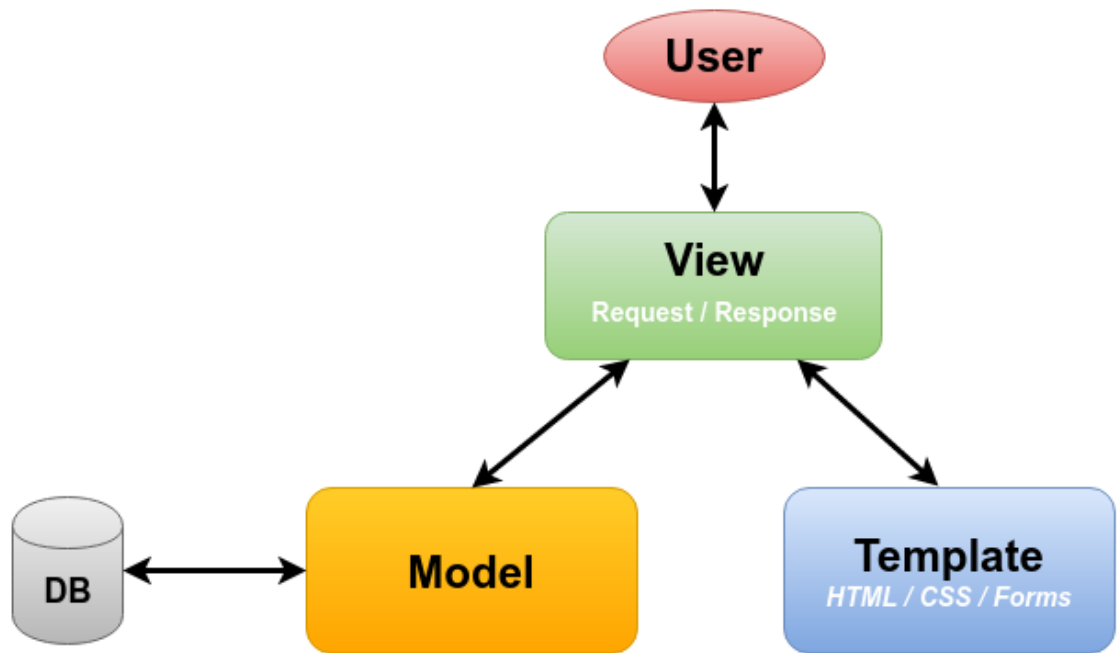


Рисунок 2.3 – Архітектура MVT

Для навчання рекомендаційної системи необхідно створити і змодельовати сховище даних.

Я вирішив зберігати дані у файлах формату csv.

Відповідно до тематики програмного забезпечення необхідно мати такі сутності:

- захід:
 - а) унікальний код заходу;
 - б) дата початку;
 - в) дата закінчення;
 - г) країна проведення;
 - д) місто проведення;
 - е) наявність часто вживаних слів;
- користувач:
 - а) унікальний код користувача;
 - б) місце проживання(країна та місто);
 - в) рік народження;
 - г) стать;

- учасники заходів:
 - а) код заходу;
 - б) коди учасників, представлені у вигляді списку;
- друзі користувача:
 - а) код користувача;
 - б) коди учасників-друзів, представлені у вигляді списку

Після опрацювання даних було отримано файл feature_train.csv дані з якого використовуються для навчання класифікаційних моделей.

- тренувальний набір:
 - а) унікальний код користувача
 - б) унікальний код заходу
 - в) кластер(тип) заходу
 - г) кількість зацікавлених користувачів
 - д) кількість зацікавлених друзів
 - е) відсоток зацікавлених друзів
 - ж) чи співпадає місце проведення заходу з місцем проживання користувача (країна, місто)
 - з) стать
 - и) вікова група
 - к) тривалість заходу
 - л) день тижня
 - м) кількість заходів відвіданих користувачем
 - н) кількість заходів цього ж типу відвіданих користувачем
 - о) чи зацікавлений користувач в заході

Більш детальний опис полів та зв'язків між файлами наведено на рисунку (Рисунок 2.4) та в таблицях (Таблиці 2.1 – 2.5)

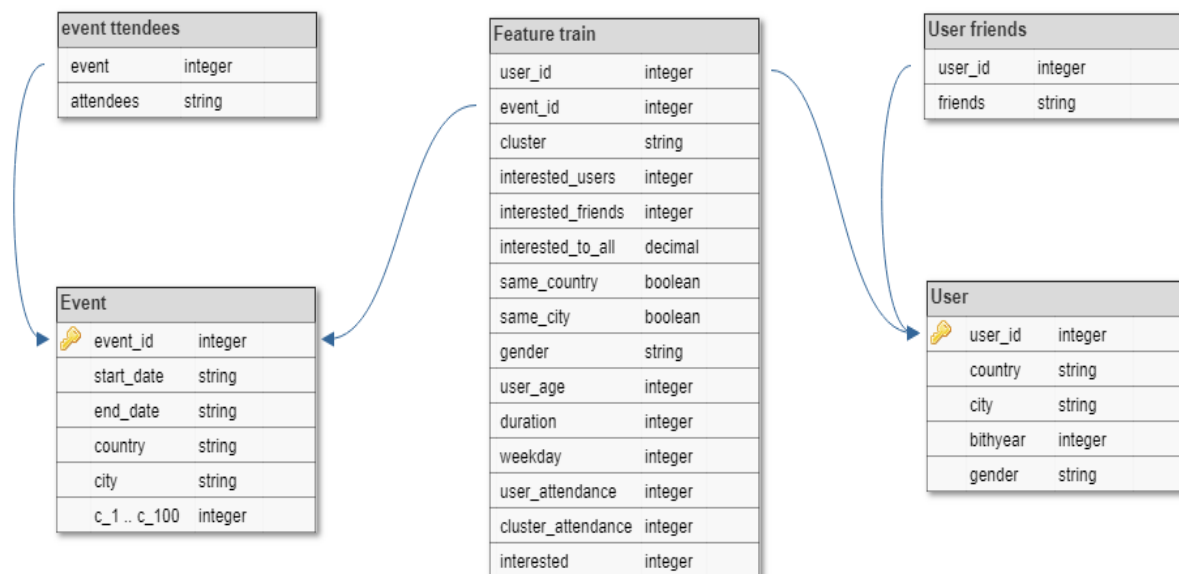


Рисунок 2.4 – Структура csv файлів

Таблиця 2.1 – Захід

Назва поля	Опис	Тип	Обов'язковість
event_id	Унікальний код заходу	Integer	Так
start_date	Дата початку заходу	String(isoformat)	Так
end_date	Дата завершення заходу	String(isoformat)	Так
country	Країна проведення	String	Ні
city	Місто проведення	String	Ні
c_1 ... c_100	Наявність часто вживаних слів	Integer	Так

Таблиця 2.2 – Користувач

Назва поля	Опис	Тип	Обов'язковість
user_id	Унікальний код заходу	Integer	Так
country	Країна в якій проживає користувач	String	Ні
city	Місто в якому проживає користувач	String	Ні
birthyear	Рік народження користувача	Integer	Ні
gender	Стать користувача	String('male' or 'female')	Ні

Таблиця 2.3 – Учасники заходів

Назва поля	Опис	Тип	Обов'язковість
event_id	Унікальний код заходу	Integer	Так
attendees	Учасники заходу	String	Ні

Таблиця 2.4 – Друзі користувача

Назва поля	Опис	Тип	Обов'язковість
user_id	Унікальний код користувача	Integer	Так
friends	Друзі користувача	String	Ні

Таблиця 2.5 – Тренувальний набір

Назва поля	Опис	Тип	Обов'язковість
user_id	Унікальний код користувача	Integer	Так
event_id	Унікальний код заходу	Integer	Так
cluster	Тип заходу	String	Так
interested_users	Кількість користувачів які відвідають захід	Integer	Так
interested_friends	Кількість друзів які відвідають захід	Integer	Так
interested_to_all	Відсоток друзів користувача, які зацікавлені в заході	Decimal	Так
same_country	Місто в якому проживає користувач	Boolean	Так
same_city	Рік народження користувача	Boolean	Так
gender	Стать користувача	String('male' or 'female')	Так
user_age	Вікова група користувача	Integer	Так
duration	Тривалість заходу	Desimal(is hours)	Так
weekday	День тижня	Integer	Так

Продовження таблиці 2.5

user_attendance	Кількість відвіданих користувачем заходів	Integer	Так
cluster_attendance	Кількість відвіданих користувачем заходів данного типу	Integer	Так
interested	Заікавленість користувача заходом	Integer	Так

2.3 Вхідні та вихідні дані

При виборі користувачем заходу, інформація про нього подається на вхід рекомендаційному модулю (таблиця 2.6).

Таблиця 2.6 – Вхідні дані

Поле	Опис	Тип
Код заходу	Унікальний код заходу	Integer
Опис	Опис заходу	String
Початок	Дата та час початку заходу	String(isoformat)
Закінчення	Дата а час завершення заходу	String(isoformat)

Продовження таблиці 2.6

Країна	Країна в якій проводиться захід	String
Місто	Місто в якому проводиться захід	String
Учасники	Список користувачів, які уже придбали або забронювали квитки	Integer

Дані про захід зіставляються з клієнтською базою і подаються на вхід навченій моделі (таблиця 2.7)

Таблиця 2.7 – Проміжні дані

Назва поля	Опис	Тип
Користувач	Унікальний код користувача	Integer
Захід	Унікальний код заходу	Integer
тип	Тип заходу	String
зацікавлені	Кількість користувачів які відвідають захід	Integer
Зацікавлені друзі	Кількість друзів які відвідають захід	Integer

Продовження таблиці 2.7

Вісоток зацікавлених друзів	Відсоток друзів користувача, які зацікавлені в заході	Decimal
Та ж країна	Місто в якому проживає користувач	Boolean
Те ж місто	Рік народження користувача	Boolean
Стать	Стать користувача	String('male' or 'female')
Вік	Вікова група користувача	Integer
Тривалість	Тривалість заходу	Desimal(is hours)
День тижня	День тижня	Integer

Для кожного рядку дата сету, модель передбачує рішення користувача. З користувачів, чия відповідь позитивна, формується список потенційних клієнтів (таблиця 2.8).

Таблиця 2.8 – Вихідні дані

Назва поля	Опис	Тип
Email	Email користувача	String
Країна	Країна в якій проживає користувач	String
Місто	Місто в якому проживає користувач	String

Продовження таблиці 2.8

Рік народження	Рік народження користувача	Integer
Стать	Стать користувача	String('male' or 'female')

Схема обробки данихображена на рисунку нижче (рисунок 2.6)



Рисунок 2.6 – Схема обробки даних

2.4 Конструювання програмного забезпечення

Перелік основних функцій з описом наведено в таблиці нижче (Таблиця 2.6).

Таблиця 2.9 – Основні функції

Назва методу	Аргументи	Значення, яке повертається	Призначення методу
preprocess_data	out_file	—	Підготувати дані для тренування моделі та записати їх в файл
generate_model_data	—	—	Розширити та валідувати тренувальні дані

prepare_data_for_event	event_data	—	Підготувати датасет для передбачення рішення користувачів щодо переданого заходу
expand_users	—	Датафрейм з інформацією про користувачів	Витягти інформацію про користувачів з csv файлу

Продовження таблиці 2.9

merge_users	data	Датафрейм з тренувальними даними	З'єднати інформацію про користувачів з переданим дата фреймом
expand_events	data	Датафрейм з тренувальними даними	Витягти інформацію про заходи з csv файлу та оєднати її з переданим дата фреймом
find_friends	data	Словник формату користувач: список друзів	Витягує інформацію про друзів користувача з файлу
event_attendees	data	Словник формату захід: учасники	Витягує інформацію про учасників заходу з файлу
get_number_of_interested_friends	row, event_attendees, user_friends	Кількість друзів користувача, які зацікавилися даним заходом	Рахує кількість друзів користувача, які зацікавилися заходом
get_number_of_interested	row, event_attendees	Кількість учасників заходу	Підраховує кількість учасників заходу

Продовження таблиці 2.9

get_user_attendance	data	Два словники. Перший формату користувач: кількість відвіданих заходів. Другий формату (користувач, кластер): кількість відвіданих заходів	Підраховує кількість відвіданих заходів для користувача. Та кількість відвіданих ним заходів з кожного кластеру.
add_friend_attendees	train_data, test_data, user_friends, event_attendees	Повертає датафрейм з тренувальними даними	Розширює тестові дані інформацією про відвідування користувачем та його друзями заходів.
fix_NA	row	Оброблений рядок датафрейму	Заповнює пропуски в переданому рядку
calculate_age_group	row	Вікова група користувача	Розраховує вікову групу для користувача в переданому рядку
get_weekday	row	День тижня	Розраховує день тижня для початку заходу

Продовження таблиці 2.9

get_clusters	data	Датафрейм з тренувальними даними	Кластеризує заходи
--------------	------	----------------------------------	--------------------

2.5 Аналіз безпеки даних

В данному програмному забезпеченні я використовую наступні засоби для захисту даних

- CSRF. Django має вбудований захист від більшості типів атак CSRF. Захист CSRF працює, перевіряючи секретність кожного запиту POST. Це гарантує, що зловмисний користувач не може просто «відтворити» форму на сайті, а зареєстрований користувач мимоволі її відправить;[4]

- Захист від SQL ін'єкцій. При виконанні запитів Django робить їх параметризацію. Це означає, що SQL код запиту відокремлюється від його параметрів. Це робить неможливим проведення SQL ін'єкцій;[4]

- Захист сесій. Кожна сесія користувача захищена унікальним JWT токеном, який містить інформацію про користувача та сесію;

2.6 Висновки по розділу

Під час написання даного розділу було розроблено діаграму бізнес-процесів з використанням нотації IDEF0. Розроблено схему сховища даних. Побудовано архітектуру програмного забезпечення, а також створено проект з використанням популярної IDE для python - PyCharm Professional 2019.1.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Щоб охарактеризувати якість програмного забезпечення було виділено наступні характеристики та підхарактеристики:

- функціональність:
 - а) придатність;
 - б) захищеність;
- надійність:
 - а) відмово стійкість;
 - б) завершеність;
- зручність використання:
 - а) зрозумілість;
 - б) вивчаємість;
- супроводжуваність:
 - а) гнучкість;
 - б) зручність тестування.

Таблиця 3.1 – Відношення впливу до якості

Хар. впливу Харак. Росту якості		Функціональність		Надійність	
		Придатність	Захищеність	Відмовостійкість	Завершеність
Функціональність	Придатність	+	-	-	+
	Захищеність	-	+	-	+
Надійність	Відмовостійкість	-	-	+	-
	Завершеність	+	-	-	+
Зручність використання	Зрозумілість	-	-	-	-
	Вивчаємість	-	-	-	-
Супроводжуваність	Гнучкість	-	---	-	-
	Зручність тестування	-	-	-	-

Продовження таблиці 3.1

Хар. впливу Харак. Росту якості		Зручність використання		Супроводжуваність	
		Зрозумілість	Вивчаємість	Гнучкість	Зручність тестування
Функціональність	Придатність	-	-	-	-
	Захищеність	-	-	---	-
Надійність	Відмовостійкість	-	-	-	-
	Завершеність	-	-	-	-
Зручність використання	Зрозумілість	+	+	+	-
	Вивчаємість	+	+	-	-
Супроводжуваність	Гнучкість	+	-	+	---
	Зручність тестування	-	-	---	+

Таблиця 3.2 - Вагові атрибути

Характеристика	Атрибут	Вага
Функціональність	Придатність	0.7
	Захищеність	1
Надійність	Відмовостійкість	0.8
	Завершеність	0.7
Зручність використання	Зрозумілість	0.5
	Вивчаємість	0.3
Супроводжуваність	Гнучкість	1
	Зручність тестування	0.8

Таблиця 3.3 - Відображення моделі зовнішньої якості на експлуатаційну якість

Хар. впливу		Експлуатаційна якість			
		Продуктивність	Задоволеність	Результативність	Безпечність
Харак. Росту якості	Функціональність	Придатність	+	-	-
		Захищеність	+	-	-
Надійність	Відмовостійкість	-	-	-	-
	Завершеність	+	-	-	-
Зручність використання	Зрозумілість	-	-	-	-
	Вивчаємість	-	+	-	-
Супроводжуваність	Гнучкість	-	-	-	-
	Зручність тестування	+	-	-	+

Таблиця 3.4–Характеристики - вага

Характеристика	Вага
Результативність	0.4
Продуктивність	0.5
Безпечність	0.6
Задоволеність	0.5

Результати тестування

- 14 – кількість функцій
- 12 – кількість відмов
- 9 – к-сть реалізованих функцій
- 6 – к-сть використовуваних користувачем функцій
- 215 – к-сть звернень користувача до системи і даних
- 202 – к-сть звернень, записаних в лог
- 4 – к-сть помилок
- 1 – к-сть вильотів
- 50 – к-сть пунктів відповідності надійності
- 35 – к-сть реалізованих пунктів відповідності надійності
- 13 – к-сть зрозумілих функцій
- 24 – к-сть змін
- 108 – Sum (A / B), A – час потрачений на зміну, B – розмір зміни
- 15 – к-сть тестів
- 12 – к-сть тестів з вбудованими функціями

Таблиця 3.5 - Модель зовнішньої якості

Характеристика	Атрибут	Вага	Формула для вимірювання	Шкала ранжування		Результат
Функціональність	придатність	0.7	$X=1 - A / B$ $A= \text{Кількість пропущених функцій}$ $B= \text{Кількість функцій описаних в специфікаціях}$	Відмінно	0.7 – 0.1	$1 - 5/14 = 0.6$
				Добре	0.5 – 0.7	
				Задовільно	0.4 – 0.5	
				Незадовільно	0 – 0.4	
	Захищеність	1	$X= A / B$ $A= \text{Кількість запитів користувачів до системи та даних, записане в лог}$ $B= \text{Кількість запитів користувачів до системи та даних, зроблених під час дослідження}$	Відмінно	0.9 - 1	$202/215 = 0.93$
				Добре	0.7 – 0.9	
				Задовільно	0.6 – 0.7	
				Незадовільно	0 – 0.6	

Продовження таблиці 3.5

Надійніс ть	Відмовостій кість	0.8	X=1-A/B А- к-сть вильотів В- к-сть помилкок	Відмінно	0.7 – 1	1 – 1 / 4 = 0.75
				Добре	0.5 – 0.7	
				Задовільно	0.4 – 0.5	
				Незадовільно	0 – 0.4	
	Завершенніс ть	0.7	X = 1 - А / В А= Кількість пунктів відповідності надійності, які не були реалізовані В= Загальна кількість пунктів відповідності надійності	Відмінно	1	1 – 14/54 = 0.74
				Добре	0.85 – 1	
				Задовільно	0.75 – 0.85	
				Незадовільно	0 – 0.75	

Продовження таблиці 3.5

Зручність застосування	Зрозумілість	0.5	$X = A / B$ $A =$ Кількість зрозумілих функцій $B =$ Загальна кількість функцій	Відмінно	1	13/14 = 0.92
				Добре	0.85 – 1	
				Задовільно	0.75 – 0.85	
				Незадовільно	0 – 0.75	
	Вивчаємість	0.1	T — Час необхідний користувачу, для вивчення функцій у хвилинах	Відмінно	< 20	6 хв = 1
				Добре	20 – 30	
				Задовільно	30 - 40	
				Незадовільно	>40	
Супроводжуваність	Гнучкість	0.6	$T = \text{Sum} (A / B) / N$ $A =$ Час роботи для змін $B =$ Розмір зміненого ПЗ $N =$ Кількість змін	Відмінно	0 - 3	4.5
				Добре	3 – 10	
				Задовільно	10 - 20	
				Незадовільно	>20	
	Зручність тестування	0.4	$X = A / B$ $A =$ Кількість тестів в яких можуть використовуватись вбудовані функції $B =$ Кількість тестів	Відмінно	0.7 – 1	12/14 = 0.85
				Добре	0.5 – 0.7	
				Задовільно	0.4 – 0.5	
				Незадовільно	0 – 0.4	

Таблиця 3.6 - Модель експлуатаційної якості

Характеристика	Атрибут	Вага	Формула для вимірювання	Шкала ранжування		Результат
Результативність	Завершеність завдань	0.4	$X = A/V$ $A = \text{к-ть завершених завдань} = 6$ $V = \text{к-ть поставлених завдань} = 9$	Відмінно	1	$6/9 = 0.67$
				Добре	0.85 – 1	
				Задовільно	0.7 - 0.85	
				Незадовільно	0 – 0.75	
Безпечність	Економічний збиток (economic damage)	0.6	$X = 1 - A/V$ $A - \text{к-сть разів нанесення економічного збитку} = 5$ $V - \text{заг к-сть випробуваних ситуацій} = 50$	Відмінно	1	0.9
				Добре	0.95 – 1	
				Задовільно	0.95 - 0.75	
				Незадовільно	0 – 0.75	
Задоволеність	Шкала задоволення	0.6	Соціологічне опитування, ле користувачі оцінюють своє задоволення в балах. Від 1 до 10	Відмінно	10	10
				Добре	8-10	
				Задовільно	7-8	
				Незадовільно	6	
Продуктивність	Витрата часу	0.5	Т = час необхідний для виконання одного завдання, у хвиликах	Відмінно	<5	6
				Добре	5 – 8	
				Задовільно	8 – 15	
				Незадовільно	>15	

Інтегральна шкала для моделі зовнішньої якості

Відмінно *верхня межа* = $1*0.7 + 1*1 + 1*0.8 + 1*0.7 + 1*0.5 + 1*0.3 + 1*0.6 + 1*0.4 = 5$

Відмінно *нижня межа* = $0,7*0.7 + 0,9*1 + 0,7*0.8 + 1*0.7 + 1*0.5 + 1*0.3 + 0,9*0.6 + 0,7*0.4 = 4,27$

					КПІ.ІП-XXXX.XXXXXXX.XX.XX	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

$$\text{Добре верхня межа} = 0,7*0,7 + 0,9*1 + 0,7*0,8 + 1*0,7 + 1*0,5 + 1*0,3 + 0,9*0,6 + 0,7*0,4 = 4,27$$

$$\text{Добре нижня межа} = 0,5*0,7 + 0,7*1 + 0,5*0,8 + 0,85*0,7 + 0,85*0,5 + 0,8*0,3 + 0,7*0,6 + 0,5*0,4 = 3,33$$

$$\text{Задовільно верхня межа} = 0,5*0,7 + 0,7*1 + 0,5*0,8 + 0,85*0,7 + 0,85*0,5 + 0,8*0,3 + 0,7*0,6 + 0,5*0,4 = 3,33$$

$$\text{Задовільно нижня межа} = 0,4*0,7 + 0,6*1 + 0,4*0,8 + 0,75*0,7 + 0,75*0,5 + 0,6*0,3 + 0,6*0,6 + 0,4*0,4 = 2,8$$

$$\text{Незадовільно верхня межа} = 0,4*0,7 + 0,6*1 + 0,4*0,8 + 0,75*0,7 + 0,75*0,5 + 0,6*0,3 + 0,6*0,6 + 0,4*0,4 = 2,8$$

$$\text{Незадовільно нижня межа} = 0*0,8 + 0*0,7 + 0*0,5 + 0*0,3 + 0*0,6 = 0$$

Таблиця 3.7 – Оцінка зовнішньої якості

Відмінно	5 – 4.27
Добре	4.27 – 3.33
Задовільно	3.33 – 2.8
Незадовільно	2.8 - 0

Інтегральна шкала для моделі експлуатаційної якості

$$\text{Відмінно верхня межа} = 1*0,4 + 1*0,6 + 1*0,6 + 1*0,5 = 2,1$$

$$\text{Відмінно нижня межа} = 1*0,4 + 1*0,6 + 1*0,6 + 1*0,5 = 2,1$$

$$\text{Добре верхня межа} = 1*0,4 + 1*0,6 + 1*0,6 + 1*0,5 = 2,1$$

$$\text{Добре нижня межа} = 0,85*0,4 + 0,95*0,6 + 0,8*0,6 + 0,8*0,5 = 1,79$$

$$\text{Задовільно верхня межа} = 0,85*0,4 + 0,95*0,6 + 0,8*0,6 + 0,8*0,5 = 1,79$$

$$\text{Задовільно нижня межа} = 0,75*0,4 + 0,75*0,6 + 0,6*0,6 + 0,6*0,5 = 1,41$$

$$\text{Незадовільно верхня межа} = 0,75*0,4 + 0,75*0,6 + 0,6*0,6 + 0,6*0,5 = 1,41$$

$$\text{Незадовільно нижня межа} = 0*0,4 + 0*0,6 + 0*0,6 + 0*0,5 = 0$$

Таблиця 3.8 – Оцінка експлуатаційної якості

Відмінно	2.1 – 2.1
Добре	2.1 – 1.79
Задовільно	1.79 – 1.41
Незадовільно	1.41- 0

Загальна інтегральна шкала

Відмінно *верхня межа* = $5 + 2.1 = 7.1$

Відмінно *нижня межа* = $4.27 + 2.1 = 6.37$

Добре *верхня межа* = $4.27 + 2.1 = 6.37$

Добре *нижня межа* = $3.33 + 1.79 = 5.12$

Задовільно *верхня межа* = $3.33 + 1.79 = 5.12$

Задовільно *нижня межа* = $2.8 + 1.41 = 4.21$

Незадовільно *верхня межа* = $2.8 + 1.41 = 4.21$

Незадовільно *нижня межа* = $0 + 0 = 0$

Таблиця 3.9 – Загальна оцінка

Відмінно	7.1 – 6.37
Добре	6.37 – 5.12
Задовільно	5.12 – 4.21
Незадовільно	4.21 – 0

Інтегральна шкала для моделі експлуатаційної якості**Отриманий результат**

$P = 4.227 + 1.847 = 6.074$ – добре

Рівень якості системи: $6.074 / 7.1 = 0.8554$.

Якість систем складає 87% від запланованої.

3.2 Опис процесів тестування

Під час тестування має бути перевірено наступне:

- а) відповідність вимогам ТЗ;
- б) коректність роботи функціональних частин;
- в) рівень безпеки даних;
- г) інтуїтивність роботи з додатком.

Методи тестування

Тестування виконується методом Gray Box Testing. Перевіряється як код, так і безпосередньо програмний продукт на відповідність функціональним вимогам. Тестування відбувається на рівні «системного тестування».

Використовуються наступні методи:

- а) базове тестування;
- б) тестування стабільності;
- в) тестування інтерфейсу.

Засоби та порядок тестування

Працездатність програмного забезпечення перевіряється шляхом:

- а) введення коректних та некоректних даних(динамічне ручне тестування);
- б) тестування функціональних вимог;
- в) статичного тестування коду;
- г) тестування інтерфейсу та зручності використання.

3.3 Опис контрольного прикладу

Результати тестування різних варіантів використання програмного забезпечення наведені в таблицях:

- авторизація (таблиця 3.10 – 3.12);
- отримання рекомендацій (таблиця 3.13 – 3.14);

					КПІ.ІП-XXXX.XXXXXXX.XX.XX	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

Таблиця 3.10 – Тест з валідними даними

Мета тесту	Перевірка авторизації з валідними даними
Початковий стан	Відкрите вікно авторизації
Вхідні дані	Логін, пароль користувача
Схема проведення тесту	Ввести в поля логіна та паролю відповідні дані. Натиснути кнопку «Вхід»
Очікуваний результат	Відкриється сторінка з заходами користувача
Стан програмного продукту після проведення випробувань	Відкрилася сторінка з заходами користувача

Таблиця 3.11 – Тест з невірним логіном

Мета тесту	Перевірка авторизації з неправильним логіном
Початковий стан	Відкрите вікно авторизації
Вхідні дані	Неправильний логін, пароль користувача
Схема проведення тесту	Ввести в поле логіну неправильне значення. Натиснути кнопку «Вхід»
Очікуваний результат	Користувач залишиться на сторінці авторизації
Стан програмного продукту після проведення випробувань	Користувач залишився на сторінці авторизації

Таблиця 3.12 – Тест з невірним паролем

Мета тесту	Перевірка авторизації з неправильним логіном
Початковий стан	Відкрите вікно авторизації
Вхідні дані	Правильний логін, неправильний пароль користувача

Продовження таблиці 3.12

Схема проведення тесту	Ввести в поле пароллю неправильне значення. Натиснути кнопку «Вхід»
Очікуваний результат	Користувач залишиться на сторінці авторизації
Стан програмного продукту після проведення випробувань	Користувач залишився на сторінці авторизації

Таблиця 3.13 – Тест з отриманням рекомендацій до заходу

Мета тесту	Перевірка отримання рекомендацій до заходу
Початковий стан	Відкрита сторінка заходами користувача
Вхідні дані	Унікальний код заходу
Схема проведення тесту	Обрати захід та натиснути кнопку «Отримати список потенційних відвідувачів»
Очікуваний результат	Завантажиться csv файл з списком потенційних відвідувачів.
Стан програмного продукту після проведення випробувань	Завантажився csv файл з списком потенційних відвідувачів.

Таблиця 3.14 – Тест з неіснуючим заходом

Мета тесту	Перевірка отримання рекомендацій до неіснуючого заходу
Початковий стан	Відкрита сторінка заходами користувача
Вхідні дані	Унікальний код заходу
Схема проведення тесту	В адресний рядок ввести неіснуючий код заходу
Очікуваний результат	Відобразиться сторінка з кодом помилки 404

Продовження таблиці 3.14

Стан програмного продукту після проведення випробувань	Відобразилась сторінка з кодом помилки 404
--	--

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Для встановлення програмного забезпечення та розгортання його на Вашому ПК необхідно клонувати репозиторій з вихідним кодом.

```
git clone https://git.teeko.io/teeko/api.git
```

Після клонування репозиторію необхідно створити віртуальне середовище з Python 3.7

```
python3.7 -m venv .venv
```

Та активувати його

```
source .venv/bin/activate
```

Далі потрібно імпортувати всі необхідні для роботи програми пакети.

```
make install
```

Щоб перевірити працездатність програми можна запустити тести

```
make test
```

Після проходження тестів Ви можете запустити програмне забезпечення на локальному сервері.

```
make run
```

4.2 Робота з програмним забезпеченням

Щоб розпочати роботу з програмним забезпеченням необхідно авторизуватись. Сторінка авторизації містить форму для вводу логіну та паролю(рисунок 4.1).

Логин:

Пароль:

[Войти](#)

Рисунок 4.1 – Форма авторизації

Після вводу необхідних для авторизації даних користувач потрапляє на сторінку зі списком заходів організатором яких він є(рисунок 4.2).

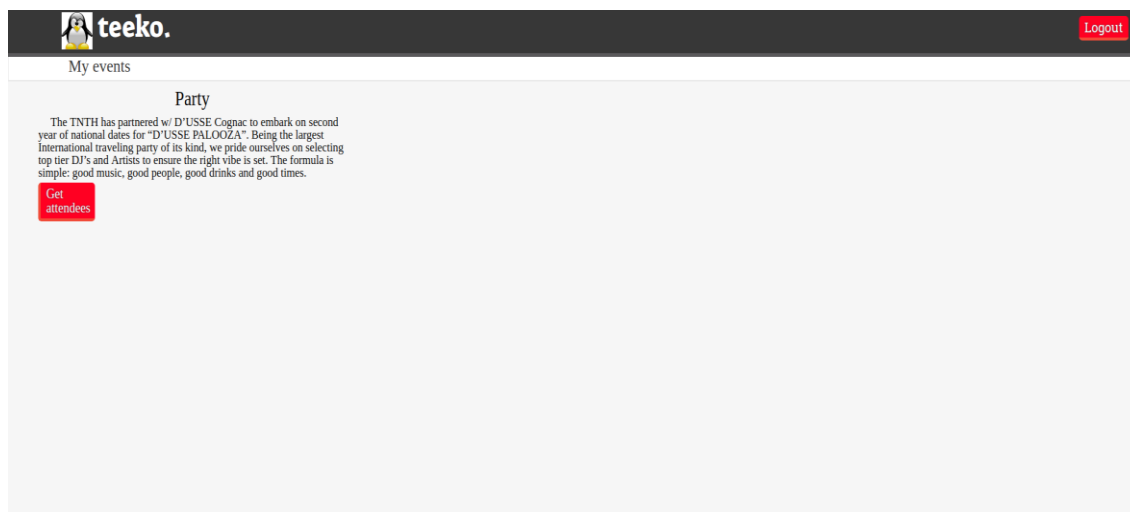


Рисунок 4.2 - Список заходів користувача

Для отримання рекомендацій необхідно обрати захід на натиснути кнопку «Отримати список»(рисунок 4.3).



Рисунок 4.3 – Кнопка «Отримати список»

					КПІ.ІП-XXXX.XXXXXXX.XX.XX	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

Після натиснення кнопки дані про захід будуть витягнені з БД, оброблені та відправлені на мікросервіс, що надає рекомендації.

Після обробки даних мікросервісом результат буде записано у csv файл та почнеться автоматичне його завантаження. Структура файлу зображена на рисунку нижче(рисунок 4.4) .

	Standard	Standard	Standard	Standard	Standard
	email	user_birthyear	user_gender	user_location	user_joinedAt
1	15342	aanderson@gibson-dennis.com	None	female	
2	16573	aandrews@stephens.com	1985	female	Phnom Penh
3	36013	aaron05@garcia.com	1982	female	
4	31547	aaron44@hotmail.com	1985	female	
5	23693	aaron58@yahoo.com	None	female	
6	7469	aaron89@yahoo.com	1982	female	Santo Domingo Dominican Republic
7	14663	aaron89@yahoo.com	1990	female	Guelph Ontario
8	3026	aaronanderson@jones.org	1979	female	Beverly Hills California
9	20167	aaroneverett@jones.com	None	female	
10	13788	aarongarcia@park-perez.com	1988	female	Bangalore India
11	19192	aarongray@decker.com	1979	female	Tehran Iran
12	11066	aarongreen@gmail.com	1987	female	San Dimas California
13	36426	aaronle@gmail.com	1978	female	Pasig
14	4231	aaronmorris@hotmail.com	None	female	
15	17253	aaronrose@gmail.com	1988	female	Phnom Penh 11
16	34770	aaronshea@yahoo.com	1984	female	Jakarta Indonesia
17	14277	aaronsnow@gmail.com	1978	female	
18	36667	abanks@cohen.com	None	female	
19	4806	abbottjennifer@gmail.com	None	female	
20	6002	abbottkevin@cortez.org	1986	female	
21	19374	aberg@gmail.com	1971	female	Toluca Mexico
22	15091	abigail88@yahoo.com	1982	female	
23	30129	abigaillewis@li.net	1985	female	Kuala Lumpur Malaysia
24	23669	abigailpatterson@hotmail.com	1986	female	
25	34018	abrewer@sanders.info	1905	male	Phnom Penh
26	17269	abrooks@yahoo.com	None	female	
27	9491	abryan@jones-brooks.com	None	female	
28	18654	abutler@torres-parrish.biz	1985	female	Hollywood California
29	6803	acamacho@hotmail.com	1988	female	Singapore Singapore
30	2324	acardenas@yahoo.com	1986	female	Huntington Park California
31	16401	acole@harris.org	1965	female	
32					

Рисунок 4.4 – Структура файлу

Після закінчення роботи з веб-сайтом необхідно вийти. Кнопка виходу знаходиться в верхньому правому кутку сторінки(рисунок 4.5).



Рисунок 4.5 – Кнопка «Вихід»

ВИСНОВКИ

У ході роботи над дипломним проектом було проаналізовано предметну область. Виділено функціональні та не функціональні вимоги. Представлена детальна архітектура проекту, а також загальний опис його математичної частини. Розроблене програмне забезпечення дало досить точні результати проте час роботи залишає бажати кращого. Отже наступним кроком буде перегляд та переосмислення існуючої архітектури проекту та деяких його функцій.

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Rajdeep D., Manpreet S., Pentreath N. Machine Learning with Spark Second Edition: навчальний посібник: 2017, 930с.
- 2) Аналіз малих даних: навчальні матеріали онлайн: [Електрон. ресурс]. – Режим доступу: <https://dyakonov.org/2016/11/14/случайный-лес-random-forest/>
- 3) Fleder D., Hosanagar K. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity: журнал: 2009, 49с.
- 4) Security in Django: онлайн документація: [Електрон. ресурс]. – Режим доступу: <https://docs.djangoproject.com/en/2.2/topics/security/>
- 5) XGBoost Documentation: онлайн документація: [Електрон. ресурс]. – Режим доступу: <https://xgboost.readthedocs.io/en/latest/>
- 6) Naive Bayes Classifier: навчальні матеріали онлайн: [Електрон. ресурс]. – Режим доступу: <https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>
- 7) Decision Tree Classifier: навчальні матеріали онлайн: [Електрон. ресурс]. – Режим доступу: http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/iguod/decisionTree.html

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації і управління

“ЗАТВЕРДЖЕНО”

В.о. завідувача кафедри

_____ О.А. Павлов

“ ____ ” _____ 2019 р.

**Програмне забезпечення формування рекомендацій з розміщення
заходів**

Графічні матеріали

КПІ.ІП-5223.045440.06.99

“ПОГОДЖЕНО”

Керівник проекту:

_____ Ю.О. Олійник

Нормоконтроль:

_____ К.І. Ліщук

Виконавець:

_____ О.А. Якимчук

Київ – 2019 року